Login Signup Lost password?

free(code):

Home Articles Browse Projects by Tag Submit new Project About Blog Help Sites Search

The Problem With Mirrors

Back to all articles

Anthony Bryan

25 Feb 2006 00:00

25Tweet

Mirrors are extremely useful when used to their full potential -- but this rarely happens. There is nothing wrong with mirrors but the way that we use them. I want to make it so average users who don't (and shouldn't need to) know too many technical details can automatically make the best use of mirrors.

As Fiber to the home (15-30 megabit speeds) and Cable/DSL (1-6 megabit speeds) become more common, some servers are having trouble maxing out a user's download pipe. One way to increase performance is to download from multiple resources at once. This is mainly useful for large files.

Mirrors are confusing to an inexperienced Web user. The Fedora Project has 110 mirror sites in North America alone. Which do you choose? Which has all the files you want? Which is quickest?

List of Fedora mirrors

In this case, not all mirrors carry all files. Some might not have all large ISOs (the Fedora Core 4 DVD image is around 2.5 gigabytes), or might only carry a subset of files (some kernel.org mirrors only have .tar.gz or .bz2 files, some have both). Or they might just be out of sync. That means you have to navigate through them to find out if they really have the file you need.

This is basically a usability problem. With some downloads, complications arise from users needing to select their Operating System, language, and location. I hope to make things easier.

Mirrors are great. We need to keep using them, but we need a better, more automatic way to use them. Peer-to-Peer (P2P) in general and BitTorrent specifically are amazing. They make it so individuals can share their bandwidth and distribute files that would otherwise cost too much through traditional server-to-client downloads.

But... P2P and regular hyperlinks are not that reliable. A hyperlink is one link to a file. If that file is gone or moved, or the server is temporarily down, that's it. 404 Error. You can search by filename, but there is no unique identifier to find that file again on the Web. P2P sharing is ephemeral. Most files are not available constantly or for the long term. I'm sure everyone has found a .torrent that he really wants, but that no one is sharing any more. BitTorrent downloads will not complete if there are no seeds at 100%. A torrent download will sit at 99.9% forever until a 100% seed (someone with the full file) starts sharing. There is no fallback plan.

I have been working on a file format called MetaLink that bundles the various methods (P2P/HTTP/FTP) of downloading files in order to improve usability, performance, reliability, and efficiency over one P2P method or a regular hyperlink. One of the main goals is to make the download process simpler for the end user. I hope this format will be found useful by Free and Open Source software projects.

Performance is increased because you download from multiple resources at the same time. Reliability is greater because there are multiple avenues or alternate locations to get a file. Hyperlinks have a single point of failure. Metalinks do not; all resources have to go out at the same time for a file to be unavailable. And it is more efficient because it spreads the downloads more evenly across multiple resources (P2P or Web/FTP servers) by multi-threading (a.k.a. segmenting or accelerating) downloads. That means that a portion of each file is downloaded from separate servers.

The minimum requirement for Metalink to be integrated into a program is that it already supports segmented downloads. Clients should also have a way to check MD5 and SHA-1 sums. And if it has BitTorrent and other P2P methods (ed2k links, magnet links, Gnutella) built in, even better. The perfect client will be able to share and access files across many P2P networks.

A few clients are implementing MetaLink right now and should be available shortly.

Here is an example MetaLink for OpenOffice.org 2.0 with links for a BitTorrent .torrent, magnet, ed2k, FTP, and HTTP. A really useful MetaLink will include combinations for different Operating Systems and languages.

```
<?xml version="1.0" encoding="UTF-8"?>
<metalink version="2.0" xmlns="http://www.m3talink.org/"
  origin="http://www.openoffice.org/mmm/OpenOffice.org-2.0.1.metalink"
  type="static" pubdate="2005-12-21-22:07:22"
refreshdate="2005-12-23-03:24:18">

<files>
  <file name="OOo_2.0.1_LinuxIntel_install.tar.gz">
    <identity>OpenOffice.org</identity>
```

```
    <version>2.0.1</version>
    <description>OpenOffice.org 2.0.1 - free office
suite</description>
    <tags>OpenOffice.org, office suite, OpenDocument, open
source</tags>
    <language>en-US</language>
    <os>Linux-x86</os>
    <size>109237237</size>
    <verification>
      <md5>e0d123e5f316bef78bfdf5a008837577</md5>
    </verification>
    <publisher>
      <name>OpenOffice.org</name>
      <url>http://www.openoffice.org/</url>
    </publisher>
    <license>
      <name>LGPL</name>
      <url>http://www.gnu.org/copyleft/lesser.html</url>
    </license>
    <copyright>Copyright 2000-2005 Sun Microsystems
Inc.</copyright>
    <resources>
      <magnet>
        <url>

magnet:?xt=urn:sha1:TWTEVOAO2IIEV67QT2ZITTXHXEUR4EXD&xt=urn:kzhash:07b7760f1c05440c779479b50dd9
OpenOffice.org_2.0.1_LinuxIntel_install.tar.gz&xs=http://ftp.snt.utwente.nl/pub/software/openof
    </url>
    <preference>90</preference>
      </magnet>
      <ed2k>
        <url>

ed2k://|file|OpenOffice.org_2.0.1_LinuxIntel_install.tar.gz|109237237|8966658D3B75FF12E1260371A
http://ftp.snt.utwente.nl/pub/software/openoffice/stable/2.0.1/OOo_2.0.1_LinuxIntel_install.tar
        </url>
    <preference>90</preference>
      </ed2k>
      <bittorrent>
    <torrent>

<url>http://borft.student.utwente.nl:6969/file?info_hash=%53%13%06%4e%30%c4%1e%e2%6f%e2%b0%24%8
        </torrent>
    <preference>100</preference>
      </bittorrent>
      <http>

<url>http://mirrors.isc.org/pub/openoffice/stable/2.0.1/OOo_2.0.1_LinuxIntel_install.tar.gz</ur
    <location>US</location>
    <preference>80</preference>
      </http>
      <ftp>

<url>ftp://ftp.ussg.iu.edu/pub/openoffice/stable/2.0.1/OOo_2.0.1_LinuxIntel_install.tar.gz</urlr
    <location>US</location>
    <preference>20</preference>
      </ftp>
      <http>

<url>http://mirrors.ibiblio.org/pub/mirrors/openoffice/stable/2.0.1/OOo_2.0.1_LinuxIntel_instal
    <location>US</location>
    <preference>20</preference>
      </http>
      <ftp>

<url>ftp://openofficeorg.secsup.org/pub/software/openoffice/stable/2.0.1/OOo_2.0.1_LinuxIntel_i
    <location>US</location>
    <preference>40</preference>
      </ftp>
```

```
    </resources>
  </file>
</files>

</metalink>
```

The goal is simplicity. A user will click this one .metalink, and the client will download the file in segments from P2P and mirrors. After the download is complete, the checksums will be compared to verify that the files are identical.

So, to sum up, these are the benefits over traditional methods:

It combines FTP and HTTP with Peer-to-peer (P2P, shared bandwidth).
It uses a standard unified format that collects links for automatic accelerated (segmented) downloads from multiple sources.
Automatic load balancing distributes traffic so individual servers are under less strain.
There's no Single Point of Failure as with FTP or HTTP URLs, so there's more fault tolerance.
There's no long, confusing list of possibly outdated mirrors and P2P links.
It makes the download process simpler for users (automatic selection of language, Operating System, location, etc.).
It stores more descriptive and useful information for Electronic Software Distribution.
There's no separate MD5/SHA-1 file or manual process for verification.
It uniquely identifies files, so even if all references to it in the Metalink stop working, the same file can be found via a P2P or Web search.
It can finish BitTorrent downloads even if no full seeds are shared.
For FTP/HTTP, an updated client is needed, but not a separate client as for P2P. (For example, the official BitTorrent client is a 6.5 megabyte download).
I'd be interested in any comments you have.

Author's bio

Anthony Bryan usually sits on his lazy bum all day, but this time he's done something. Luckily, that something doesn't involve physical movement, but it may allow him to get a new chair sometime in the next five years. Probably... Possible improvements to the download process -- by an otherwise lazy bum.

Editorials

Related projects

BitTorrent

Recent comments

25 Feb 2006 07:45 LX

> Which clients are implementing the standard?
> First, you mentioned clients to implement this new standard. Which ones?
>
> Second, there ought to be a nice little utility to create such metalinks (as most people are too lazy to remember all those xml tags or even type them).
>
> Otherwise, this is a great idea - should do a good job on download acceleration, too!
>
> Greetings, LX

25 Feb 2006 10:40 gustafg

> Good idea, but implementation raises questionmarks
> I think the idea behind this is plausible but I wonder if all the assumptions are correct, these are my questions/reservations etc:
>
> The mirror problem, there is nothing that prevents a large site from verifying its mirrors and update its web site dynamically. There is nothing from preventing them to dynamically only present a subset of all mirrors at any given time and by doing so creating a form of load sharing. Even if this would be a site specific implementation it could work similar to how multiple dns records work to ease load on large internet sites. In fact, if you could get your http/ftp mirrors to agree on a common directory structure you could create the loadsharing this way for downloads only.
>
> The P2P (read BitTorrent) problem and the no seeds argument is pretty much void for anyone distributing their own content in this way. If I choose to distribute my project via BitTorrent I of course ensure that I myself is always seeding.
>
> Another problem is that in order for segmented downloads to work you put a lot of pressure on client implementations. I cannot see how you could possibly successfully mix a BitTorrent download and a FTP download unless the client itself implements both of these protocols.
>
> Servers need to support segmented uploads, at least not all FTP servers do as far as my knowledge is correct. Clients needs to handle this as well.
>
> The single point of failure argument is only true if the site serving the metalink itself is redundant, not having access to the metalink is just as much a problem as broken mirrors are.

It seems the proposed solution is a quite complex and therefor I remain skeptical about its success.

I also have some suggestions for you.

You may want to include a preference parameter between different protocols, as I understand it now the preference parameter is used only to choose between mirrors of same type.

You should start developing a metalink library in various languages to be used for interpreting these links aswell as doing the downloading. This way it seems to me client acceptance would be easier to achieve.

Above is unless you intend to actually create and distribute a metalink client which could be launched for instance by a web browser when it downloads a given metalink.

Anyways, its nice to see new refreshing ideas :-)

25 Feb 2006 13:01 answerguy

“ Round Robin DNS + Virtual Hosting (+ optional BGP Virtual IP Routing)
It's possible to provide mirror transparently through a combination of methods. The easiest is round robin DNS with web/ftp virtual hosting. This is basically how the Debian archives scale.

A more advanced technique can be used among (or with the co-operation of) BGP peering customers (obviously requires an AS number, etc). In this technique you configure a single virtual IP address (per "service") on each mirror node. Then you propagate your routes to this VIP using the normal BGP4 Internet infrastructure.

To the routing tables these all look like different routes to one machine. (The fact that they actually exist on multiple machine in diverse locations is irrelevant to the upper layer protocols so long as the contents and services provided or synchronized via some out-of-band method --- such as the "real" IP addresses of the mirror hosts).

The huge advantage of this sort of BGP/VIP method is that each client is transparently routed to their "closest" mirror (along the most efficient route).

I read that Nominum.net (developers of the BIND9 updates to the canonical/reference implementation of the DNS standards) used this technique for their DNS load balancing.

(A similar technique should work for intranet applications over any good dynamic routing protocol such as OSPF).

Unfortunately I don't know of any RFCs or detailed technical articles spelling out all the details. All I have is the conceptual overview gleaned from chatting at some geekfest (probably over brews).

JimD

25 Feb 2006 13:41 kodekrash

“ XML Structure
For my own education, I'm writing a metalink parser/generator in PHP. I'm going to make a database of metalinks for all the packages in the Fedora YUM repository as a test, and I've run into a couple things...

I can see that you've put some work into the XML vocabulary, but it seems ill-suited for efficient parsing. I have two specific elements in mind:

<verification> and <resources>

In the verification element, you use <md5> as a sub-element. I assume this is because you plan to have multiple verification methods, for example, let's add an SHA1 option:

<verification>

<md5>[hash]</md5>

<sha1>[hash]</sha1>

</verification>

This means that a parser must look for 2 different element names, even though the element is the same thing - a hash type and key.

A more efficient method might be something like this:

<verification>

<hash type="md5">[hash]</hash>

<hash type="sha1">[hash]</hash>

</verification>

With this, a parser can very simply parse all the verificiation options with a simple loop for each <hash /> element.

Same thing for <resources>, where you use the protocol name as the element, such as <magnet>.

Again, it would be more efficient to do something like:

<resource>

<type>magnet</type>

<url>magnet:[uri]</url>

<preference>90</preference>

</resource>

instead of:

<magnet>

<url>magnet:[uri]</url>

<preference>90</preference>

</magnet>

Just a couple thoughts....

26 Feb 2006 02:27 mastermitch

Could be done with BitTorrent alone
Instead of mixing HTTP, FTP and Torrents, one could just use Torrents to get the listed benefits: Torrents let you address multiple trackers, so there is no single point of failure at that point. Instead of having 5 HTTP or FTP Mirrors, you can deploy 5 "always on" seeds for your data on different hosts. That way, everyone has the chance to always reach a 100% seed. I don't see why HTTP and FTP should be added to the mix, they just make things more complicated IMHO.

Regards,

Christian

26 Feb 2006 23:27 imipak

Bandwidth management
The easiest way to pick a mirror according to resources would be to use bing or pchar to determine the available bandwidth between client and each server, then go for the one with the greatest available bandwidth.

<p>

(Latency - usually in the order of seconds - is irrelevent for a transfer that can take minutes or hours. Geography is irrelevent if the nearest has more users than capacity. Round-robin only works if both servers and clients are evenly distributed by bandwidth, which is almost certainly never the case.)

06 Mar 2006 01:51 ulricerikssson

Re: Round Robin DNS + Virtual Hosting (+ optional BGP Virtual IP Routing)

>
> A more advanced technique can be used
> among (or with the co-operation of) BGP
> peering customers (obviously requires an
> AS number, etc). In this technique you
> configure a single virtual IP address
> (per "service") on each mirror
> node. Then you propagate your routes to
> this VIP using the normal BGP4 Internet
> infrastructure.

This is unsuitable for long-lived connections, because
routing changes can suddenly direct a user to a different server in the middle of a download.

It's fine for DNS though.

06 Mar 2006 14:32 manuel_subredu

simba
I agree with you. Most of the mirrors are not transparent. You don't even know what is excluded from a mirror. You don't know when was last updated, or what the mirror size is or (worse) what was transfered on the last update. What about some rss feeds ? Do you think they are usefull ? If you do, take a look at RoEduNet Iasi Online Archive (ftp.iasi.roedu.net/mir...) . The guys from RoEduNet Iasi are using simba (simba.packages.ro) to manage their mirrors, and as you can see, almost all the information related to a mirror is available online ;)

14 Mar 2006 14:46 bishopolis

critics and salesmen
when a critic attempts to sell their own solution, it taints the critique.

It also sounds a bit like an infomercial.

It's unfortunate, for I was going with it up to the point where the selling began.

22 Mar 2006 17:42 tomkins

Re: Could be done with BitTorrent alone

> Instead of mixing HTTP, FTP and

> Torrents, one could just use Torrents to

> get the listed benefits: Torrents let

> you address multiple trackers, so there

> is no single point of failure at that

> point. Instead of having 5 HTTP or

> FTP Mirrors, you can deploy 5

> "always on" seeds for your

> data on different hosts. That way,

> everyone has the chance to always reach

> a 100% seed. I don't see why HTTP and

> FTP should be added to the mix, they

> just make things more complicated IMHO.

You could also modify the tracker to only give the IP addresses of seeds instead of any other peers. Although this sort of defeats the point of BitTorrent, it's a quick and easy solution which would solve the problems in the article by using different sources to download from.

29 Mar 2006 15:26 antini

Update
We have a site up for the project at www.metalinker.org/ (www.metalinker.org/).

If you are on Windows, you can try some of the samples on the Metalink site (www.metalinker.org/sam...) with GetRight 6 Beta (www.getright.com/beta6...).

The next version (.5.9.994?) of FlashGot (www.flashgot.net/) (cross platform Firefox extension) should also support it. There are also a few other clients adding native support.

07 Apr 2006 12:32 gw

SMTM?
Oh, and where's the price tag?

02 May 2006 02:34 CrazyGFreak

Re: Round Robin DNS + Virtual Hosting (+ optional BGP Virtual IP Routing)

>
>
> %
> % A more advanced technique can be used
> % among (or with the co-operation of)
> BGP
> % peering customers (obviously requires
> an
> % AS number, etc). In this technique
> you
> % configure a single virtual IP address
> % (per "service") on each
> mirror
> % node. Then you propagate your routes
> to
> % this VIP using the normal BGP4
> Internet
> % infrastructure.
>
>
>
> This is unsuitable for long-lived

> connections, because
> routing changes can suddenly direct a
> user to a different server in the middle
> of a download.
>
> It's fine for DNS though.
>

so does anybody know, (www-einkaufen.de/) which clients are implementing the standard? Meta links sound real nice.

02 May 2006 04:15 ulricerikssON

Re: Round Robin DNS + Virtual Hosting (+ optional BGP Virtual IP Routing)

> so does anybody know, which clients are

> implementing the standard? Meta links

> sound real nice.

Meta links, at least as described here, are IMHO a complex solution to a problem that is already solved by Bittorrent.

06 May 2006 14:19 antini

FlashGot support for Metalink
FlashGot 0.5.9.995 (www.flashgot.net) (Firefox extension) now supports an earlier version of Metalink (www.metalinker.org) with GetRight. FlashGot could be modified so Metalink could work with any of the other cross platform download managers it supports.

08 Jun 2006 17:56 antini

GetRight 6
GetRight 6 (www.getright.com/) (final version) is now out. It supports metalinks and works with Wine on Linux. I'd still love to see a command line metalink client for unix.

11 Jun 2006 21:18 antini

Re: Updated metalinks for various files
Metalink @ Packages Resources (metalink.packages.ro/) provides updated Metalinks for the Linux Kernel, OpenOffice.org, & Fedora with more Open Source projects on the way (KDE, Debian, Ubuntu, Mandriva). Software and (GPL'd) source code for generating Metalinks is also available there.

04 Jul 2006 18:29 antini

aria2 - Unix client
aria2 (aria2.sourceforge.net/) is a command line client for Unix that supports Metalink (HTTP/FTP) and BitTorrent.

09 Jul 2006 21:27 antini

OpenOffice.org uses metalinks
OpenOffice.org (distribution.openoffic...) uses metalinks.

Clients:

Mac GUI - in beta testing

Unix CLI - aria2 (aria2.sourceforge.net/)

Windows - GetRight 6 (www.getright.com/)

08 Aug 2006 17:00 antini

New and updated Metalink clients
wxDownload Fast (dfast.sourceforge.net/) is a download manager on Mac, Unix, and Windows that supports Metalink.

aria2 (aria2.sourceforge.net/) is a unix command line download utility that supports BitTorrent and Metalink. Version 0.7.0 offers updated Metalink support.

BLAG (www.blagblagblag.org/d...) offers their Linux distribution ISO for download with Metalink.

14 Aug 2006 23:32 Mark8

Thank you
Great advice, thank you!

07 Sep 2006 16:52 antini

Re: New and updated Metalink clients
Speed Download (www.yazsoft.com) (Mac) now supports Metalinks. It looks and works great, check it out.

12 Sep 2006 22:03 antini

BSD/Linux Distributions using Metalink
DesktopBSD (desktopbsd.net/), BLAG Linux (www.blagblagblag.org/d...), StartCom Linux (linux.startcom.org/), Berry Linux (yui.mine.nu/berry/edow...), Ubuntu Christian Edition (www.christianubuntu.com/)

22 Oct 2006 11:52 antini

Metalink tools
Bram Nejit has released Metalink tools (prog.infosnel.nl/metal...) which are extremely useful for making metalinks, by generating many different checksums and importing mirror lists.

06 Sep 2007 21:09 RobertGoretsky

Setting the Preference Parameter On The Server?
I understand that the metalink configuration provides a 'preference' parameter for each link that determines how likely the client should be to select that particular link. I assume that this parameter would not be static, but rather would be dynamically set by the web server providing the metalink. But how would the server know how to set this? It seems that you may lose some of the intuitive "I live near X, so I will choose the server near X" functionality you get with regular mirror hyperlinks. Your thoughts on this?

Robert H. Goretsky

Hoboken, NJ

Project Spotlight
Luksus
A drive encryption script that makes encryption of storage media and virtual files quick and easy.

Project Spotlight
No Screenshot Provided
dht
A Bittorrent DHT library.

About Freecode Privacy Cookies/Opt Out Terms Advertise Contact Us

FOLLOW US ON twitter

Copyright © 2014 Dice. All Rights Reserved. Freecode is a DiceHoldings, Inc. service.